. JOINT INTEROPERABILITY & ENGINEERING ORGANIZATION

CENTER FOR APPLICATIONS ENGINEERING
COMMAND AND CONTROL APPLICATIONS DEPARTMENT

SOFTWARE VERSION DESCRIPTION (SVD)

FOR THE

AIRFIELDS SYSTEM

Version 2.1.2

Revised 05 November 1996

SUBMITTED BY:

LCDR JOANNE OLCOTT
Chief, Command & Control
Applications Team

APPROVED BY:

LTC JAMES ROBINSON
Chief, Command & Control
Applications Department

Copies of this document may be obtained from:

Command & Control Applications Department
Attn: Code JEKE
5600 Columbia Pike
Falls Church, VA 22041

ACKNOWLEDGMENT

This document was prepared for the Defense Information Systems Agency (DISA), Joint Interoperability and Engineering Organization (JIEO), Center for Operational Support (CFOS), General Applications Department (JEKG).

This Software Version Description (SVD) contains all the information necessary to release, track, and control the re-hosted/re-engineered version of the Airfields system.

Any questions, comments, or considerations relative to this Software Version Description should be directed to the following:

Global Command and Control System (GCCS) Hotline
DSN: 653-8681
Commercial: (703) 735-8681

CONTENTS

SECTION 1. SCOPE

1.1 Identification. The Airfields system provides the Worldwide Military Command and Control System (WWMCCS) community with a wide range of data about free world airfields. All data is supplied by the Defense Mapping Agency Aerospace Center (DMAAC) and is updated monthly. The Airfields System was re-engineered from COBOL to the Ada 95 programming language and provides the capability to print the One-Line, One-Page Summary, Multi-Page, Selective Data Retrieval, and Turnaround reports both on- and off-line.

1.2 System Overview. The functional proponent for Airfields is the Joint Staff Logistics Directorate (J4). The office of primary responsibility (OPR) is the Computer Systems Engineering/Applications Engineering Facility. The designated development Agency (DDA) is the Center for Operational Support, General Applications Department (JEKG).

The original system was hosted on proprietary hardware under a WWMCCS COBOL environment. The system's original release date was 15 May 1990. Historically, WWMCCS users access the system approximately 100 times per month. The database is owned by the Defense Mapping Agency Aerospace Center (DMAAC) and contains data on approximately 44,000 airfields and consists of over one million records.

This version of the Airfields system runs under Solaris 2.3 and utilizes a relational database management system (RDBMS) as a data manipulation tool. It is written in the Ada 95 programming language and utilizes a Graphical User Interface (GUI) at the front end of the system.

The following retrieval capabilities are included in this version:

a. One-Line Summary Report
b. One-Page Summary Report
c. Turnaround Report/Coordinate/Radius Calculation
d. Multi-Page Report
e. Selective Data Retrieval

The system complies with GCCS' Integration Standards and employs many standards such as the windowing capability and an extensive help facility to aid the user with system operation. The primary operational sites include the Global Command and Control System (GCCS) community and the Joint Staff.

1.3 Document Overview. The purpose of this Software Version Description (SVD) is to provide the information needed to release, track, and control this version of the Airfields system.

SECTION 2. REFERENCED DOCUMENTS

a. Department of Defense, Military Standard Software Development and Documentation, MIL-STD-498, 5 Dec 1994

b. Data Item Description (DID) number DI-IPSC-81442, Software Version Description (SVD), 5 DEC 1994

c. Joint Interoperability & Engineering Organization (JIEO), Washington, DC, Software Development Plan (SDP) (Draft), 28 February 1995

SECTION 3. VERSION DESCRIPTION

3.1 General. When assigning version numbers to the Airfields software, the degree of change was the determining factor used. Major changes to the software were assigned a whole number such as 2.0 while minor changes were assigned a number such as 2.3. Very minor changes to the software were assigned a number such as 2.03.

3.1.1 Description of Versions. The following are descriptions of the various versions of the Airfields System software:

a. Airfields Version 1.0.2 represents the Airfields software segment. It provides reports from the Airfields database. The current capabilities are for the one-line report by:

1. Country Code
2. Basic Encyclopedia Number
3. ICAO/FAA Code
4. GEOLOC
5. Airfield Name

b. Airfields Version 1.0.3 represents the addition of the One-Page report and the Coordinate Radius/Country Code search.

c. Version 1.0.4 represents the implementation of the Turnaround Calculator and the File/Print functions. The print function prints the reports to the system default printer. This version runs against the airfields database version 1.0.3.

Load AIRFDB 1.0.3
Load AirFields 1.0.4

d. Version 1.1 represents the implementation of the Multi-Page report under the name "Detail Report." The Selective Data Retrieval and the "Help" function (which displays the User Manual) were also implemented under this release. For instructions on how to install and de-install this segment, refer to the following:

1. GCCS Implementation Procedures for AIC
   GCCS V2.1, Route 0, final
   Dated 09/27/95
   CM# LL-500-103-18

2. GCCS System Administration Manual
   Route 0, final
   Dated 09/29/95
   CM# LL-500-29-10

e. Version 2.0 represents the full implementation of the Airfields Application including license management, full reporting capability, print, Help (rudimentary). Errors in the Coordinate-Radius processing were also fixed in this release. For instructions on how to install and de-install this segment, refer to the following:

(1) GCCS Implementation Procedures for AIC
GCCS V2.1, Route 0, final
Dated 09/27/95
CM# LL-500-103-18

(2) GCCS System Administration Manual
Route 0, final
Dated 09/29/95
CM# LL-500-29-10

f. Version 2.0.1 represents the correction of one character in the S99airfields script.

g. In Version 2.0.2, report names were reset to One-Page and Multi-Page. The Multi-Page report modules were modified to improve retrieval efficiency. An updated User Manual was also included in this release. There are no changes to the installation and de-installation instructions, however a note has been added to check license/database for problems before notifying programmers if the application aborts. The airlaunch script was modified to change directory to the /h/AirFields/progs path to allow Airfields to find the User_Manual file. The User_Manual was also updated.

h. Version 2.0.3 sets the common criteria off for all reports where the major selection criteria is BE Number. The only exception is the Security Classification Selection. Complete remarks are display/printed in the Multi-Page Report. Taxiway width in the Turnaround Calculator Report now shows "Unk" when there is no data in the Taxiway table for the airfield. There are no changes to the installation/de-installation instructions.

i. In Version 2.0.4 changes are made to resolve the GSPR G60498. The one_Line_Report with Coordinates/Radius can now accept any other values or defaults for the fields Runway Length (Min, Max), Runway Width (Min, Max), Taxiway Width (Min, Max), Load Class Nbr (Min, Max) or Surface Type.

j. In Version 2.1, 2.1.1 & 2.1.2 new Ada_Sql_Connect Software is included to replace the old license format where Flex_LM software was required to manage the license, with the new software no license is required. Also changes are made to resolve the airfield_Reports print problem & Help Problem.

3.2 Inventory of Materials Released. The following are software files, COTS, data tables/files, etc. which must be installed in order for the software to operate:

Oracle/SQL Version 7	-	Unclassified
Ada Run-Time Version 3.01	-	Unclassified
GNAT Compiler Version 3.01	-	Unclassified
Screen Machine (GUI)	-	Unclassified
Open Database Connectivity from OIS Version 2.0	-	Unclassified
Solaris Version 2.3	-	Unclassified
Airfields Database Tables Version 2.0	-	Secret/NOFORN
Airfields Executable Version 2.1.1	-	Unclassified

3.2.1 Documentation. The following is a list of Version 2.0.1 documentation that is being delivered with Version 2.0.1 of the Airfields system:

a. Configuration Management Plan (CM Plan) (Draft)
b. Database Design Description (DBDD) (Draft)
c. Software Development Plan (SDP) (Draft)
d. Software Requirements Specification (SRS) (Draft)
e. Software Maintenance Manual (SMM) (Draft)
f. Software User Manual (SUM) (Draft)
g. Software Version Description (SVD) (Draft)
h. Software Installation Plan (SIP) (Draft)
I. Software Center Operator Manual (SCOM) (Draft)
j. Software Test Plan (STP) (Draft)
k. Software Test Description (STD) (Draft)
l. Software Test Report (STR) (Draft)
m. GCCS Delivery Letter
n. Airfields Releasibility Rules
o. Required License Commercial Off-the-shelf (COTS) software

3.3 Inventory of Software Contents. See Section 3.3.1 - Application Files for a list of package bodies and specifications used in the system. The version number of these files is 2.0.2 and all are unclassified.

3.3.1 Application Files. The following are package bodies and specifications used in the development of the system. Each package is followed by an explanation of what the file does.

about_panel_dialog_class-interact.adb -- A separate procedure in the About Panel Dialog Class package which displays the About Panel and provides the interface to the developers' code. The About Panel displays the name, version number and date, and credits of the Airfields System. This is a Screen Machine artifact, modified to call other (developer) modules.

about_panel_dialog_class.adb -- Package body which defines procedures and functions to perform basic interaction with the About_Panel. The About Panel displays the name, version number and date, and credits of the Airfields System. This is a Screen Machine artifact.

airfield_statuses.adb -- Package body which contains functions and procedures to convert the airfield status type into a string and back into a status type, and to convert an airfield status code (from the database) to a status type or to a string.

airfield_statuses.ads -- Package specification defining the airfield status type and the interface for the functions and procedures for type conversion.

airfields.adb - The main procedure for the Airfields System. This procedure initializes the screens, displays the About Panel, then displays the Main Panel for the body of processing. When the Main Panel is closed, this procedure closes the screens and terminates processing.

airfields_object_factory_class-about_panel.adb -- This package body provides the initialization of the About Panel required by About_Panel_Dialog_Class. The specification for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class-cap_list_panel.adb -- This package body provides the initialization of the Cap List Panel required by Cap_List_Panel_Dialog_Class. The specification for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class-help_panel.adb -- This package body provides the initialization of the Help Panel required by Help_Panel_Dialog_Class. The specification for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class-main_panel.adb -- This package body provides the initialization of the Main Panel required by Main_Panel_Dialog_Class. The specification

for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class-selection_criteria_panel.adb -- This package body provides the initialization of the Selection Criteria Panel required by *selection_criteria_panel_dialog_class*. The specification for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class-selective_panel.adb -- This package body provides the initialization of the Selective Panel which is required by Selective_Panel_Dialog_Class. The specification for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class-turnaround_panel.adb -- This package body provides the initialization of the About Panel which is required by About_Panel_Dialog_Class. The specification for this package may be found in Airfields_Object_Factory_Class.ads. This is a Screen Machine artifact.

airfields_object_factory_class.adb -- This package body contains the functions that create and return the user interface panel data structures. This is a Screen Machine Artifact.

airfields_object_factory_class.ads -- This package specification contains the field name to field number mapping constant definitions and the functions that create and return the user interface panel data structures. This is a Screen Machine artifact.

cap_list_panel_dialog_class-interact.adb -- A separate procedure in the Cap List Panel Dialog Class package which displays the Cap List Panel and provides the interface to the developers' code. The Cap List Panel displays a list box with a selection of possible Aircraft Capacities for user selection. This is a Screen Machine artifact, modified by developer.

cap_list_panel_dialog_class.adb -- Package body which defines procedures and functions to perform basic interaction with the Cap_List_Panel. The Cap List Panel displays a list box with a selection of possible Aircraft Capacities for user selection. This is a Screen Machine artifact.

cap_list_panel_dialog_class.ads -- Package specification which defines procedures and functions to perform basic interaction with the Cap_List_Panel. The Cap List Panel displays a list box with a selection of possible Aircraft Capacities for user selection. This is a Screen Machine artifact.

center_string.adb -- a function which positions a string at the center of the printed line.

check_for_digit.adb -- This procedure reads in a string and checks to see if the first character is a number. If the condition is true (i.e., it is a number), it concatenates the string and puts a single quote at the beginning and end of that string. If the condition is false, then the original string is passed back.

debug.adb -- This package body is part of a GNAT debug package/tool used to turn debug on or off in a program.

debug.ads -- This package specification is part of a GNAT debug package/tool used to turn debug on or off in a program.

debug_pkg.adb -- A Package body holding the procedures to detect the state of the debug flag which allows the output of Text-io statements to the screen if the flag is set on.

debug_pkg.ads -- A package specification defining the debug flag and providing the declarations of the functions and procedures needed to put debugging statements to the screen.

dm_airport-bind_all.adb -- This package body is a separate procedure which binds the SQL record with ODBC. It is being called by the dm_airport data manager to create SQL and ODBC binding.

dm_airport-convert_to_datavalue.adb -- This package body is a separate function called by the fuel_dispensing data manager to assign the retrieved values from the database to the return variables.

dm_airport-create_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the dm_airport data manager based on the user supplied key.

dm_airport-dm_acft_bunkers-bind_acft_bunkers.adb -- This package body is a separate procedure which binds the SQL record with ODBC. It is being called by the acft_bunkers data manager to create SQL and ODBC binding.

dm_airport-dm_acft_bunkers-convert_to_acft_bunkers_data.adb -- This package body is a separate function called by the acft_bunkers data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_acft_bunkers-create_acft_bunkers_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the acft_bunkers data manager based on the user supplied key.

dm_airport-dm_acft_bunkers.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_acft_bunkers.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_apron-bind_fuel_storage.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by the apron data manager to create SQL and ODBC binding.

dm_airport-dm_apron-convert_to_apron_data.adb -- This package body is a separate function called by the dm_apron data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_apron-create_apron_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the apron data manager based on the user supplied key.

dm_airport-dm_apron.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_apron.ads -- This package specification controls the access to the *primary runway record.*

dm_airport-dm_arrest_sys-bind_arrest_sys.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by the dm_arrest_sys data manager to create SQL and ODBC binding.

dm_airport-dm_arrest_sys-convert_to_arrest_sys_data.adb -- This package body is a separate function called by the dm_arrest_sys data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_arrest_sys-create_arrest_sys_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the dm_arrest_sys data manager based on the user supplied key.

dm_airport-dm_arrest_sys.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_arrest_sys.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_country_cd.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_country_cd.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_defueling.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_defueling.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_fuel_dispensing-bind_fuel_dispensing.adb -- This package body is a separate procedure which binds the SQL record with ODBC. It is being called by the fuel dispensing data manager to create SQL and ODBC binding.

dm_airport-dm_fuel_dispensing-convert_to_fuel_dispensing_data.adb -- This package body is a separate function called by the fuel_dispensing data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_fuel_dispensing-create_fuel_dispensing_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the fuel_dispensing data manager based on the user supplied key.

dm_airport-dm_fuel_dispensing.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_fuel_dispensing.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_fuel_stock-bind_fuel_stock.adb -- This package body is a separate procedure which binds the SQL record with ODBC. It is being called by the fuel_stock data manager to create SQL and ODBC binding.

dm_airport-dm_fuel_stock-convert_to_fuel_stock_data.adb -- This package body is a separate function called by the fuel_stock data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_fuel_stock-create_fuel_stock_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the fuel_stock data manager based on the user supplied key.

dm_airport-dm_fuel_stock.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_fuel_stock.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_fuel_storage-bind_fuel_storage.adb -- This package body is a separate

procedure which binds the SQL record with ODBC. This procedure is being called by the fuel_storage data manager to create SQL and ODBC binding.

dm_airport-dm_fuel_storage-convert_to_fuel_storage_data.adb -- This package body is a separate function called by the fuel_storage data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_fuel_storage-create_fuel_storage_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the fuel_storage data manager based on the user supplied key.

dm_airport-dm_fuel_storage.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_fuel_storage.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_hangars-bind_hangars.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by the hangars data manager to create SQL and ODBC binding.

dm_airport-dm_hangars-convert_to_hangars_data.adb -- This package body is a separate function called by the hangars data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_hangars-create_hangars_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the hangars data manager based on the user supplied key.

dm_airport-dm_hangars.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_hangars.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_hardstand-bind_dm_hardstand.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by the hardstand data manager to create SQL and ODBC bindings.

dm_airport-dm_hardstand-convert_to_hardstand_data.adb -- This package body is a separate function called by the hardstand data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_hardstand-create_hardstand_sql_statement.adb -- This package body is a separate function to create an SQL statement for the hardstand data manager based on the user supplied key.

dm_airport-dm_hardstand.adb -- This package body retrieves the Hardstand records by calling the CREATE, BIND, and CONVERT packages. It is a child package to the DM_Airport package and interfaces with Open Database Connectivity (ODBC) to allow applications to access data in the database management system (DBMS) using Structured Query Language (SQL).

dm_airport-dm_hardstand.ads -- This package specification declares the database record layout to retrieve the Hardstand records. It also defines the interface to other programs.

dm_airport-dm_obf_storage-bind_obf_storage.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by every data manager to create SQL and ODBC binding.

dm_airport-dm_obf_storage-convert_to_obf_storage_data.adb -- This package body is a separate function called by every data manager to assign the retrieved values from database to the return variables.

dm_airport-dm_obf_storage-create_obf_storage_sql_statement.adb -- This package body is a separate function called by Data Managers to create an SQL statement for a particular data manager based on the given key.

dm_airport-dm_obf_storage.adb -- This package body retrieves the Obf_Storage records by calling three separate packages (CREATE, BIND and CONVERT). It is a child package to the DM_Airport package and interfaces with Open Database Connectivity (ODBC) to allow applications to access data in the database management system (DBMS) using Structured Query Language (SQL).

dm_airport-dm_obf_storage.ads -- This package specification declares the database record layout to retrieve the Obf_Storage records. It also defines the interface to other programs.

dm_airport-dm_oconus_airport-bind_oconus_airport.adb -- It is a separate procedure which binds the SQL record with ODBC. This procedure is being called by every data manager to create SQL and ODBC binding.

dm_airport-dm_oconus_airport-convert_to_oconus_airport_data.adb -- This package body is a separate function called by every data manager to assign the retrieved values from the database to the returned variables.

dm_airport-dm_oconus_airport-create_oconus_airport_sql_statement.adb -- This package body is a separate function called by Data Managers to create an SQL statement for a particular data manager based on the given key.

dm_airport-dm_oconus_airport.adb -- This package body retrieves the Oconus_Airport records by calling three separate packages (CREATE, BIND and CONVERT) and is a child package to the DM_Airport package. It also interfaces with Open Database Connectivity (ODBC) to allow applications to access data in the database management system (DBMS) using Structured Query Language (SQL).

dm_airport-dm_oconus_airport.ads -- This package specification declares the database record layout to retrieve the Oconus_Airport records. It also defines the interface to other programs.

dm_airport-dm_oconus_runway-bind_oconus_runway.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by every data manager to create SQL and ODBC binding.

dm_airport-dm_oconus_runway-convert_to_oconus_runway_data.adb -- This package body is a separate function called by every data manager to assign the retrieved values from database to the return variables.

dm_airport-dm_oconus_runway-create_oconus_runway_sql_statement.adb -- This package body is a separate function called by Data Managers to create a SQL statement for a particular data manager based on the given key.

dm_airport-dm_oconus_runway.adb -- This package body retrieves the Oconus_Runway records by calling three separate packages (CREATE, BIND and CONVERT) and is a child package to the DM_Airport package. It also interfaces with Open Database Connectivity (ODBC) to allow applications to access data in the database management system (DBMS) using Structured Query Language (SQL).

dm_airport-dm_oconus_runway.ads -- This package specification declares the database record layout to retrieve the oconus_runway records. It also defines the interface to other programs.

dm_airport-dm_oconus_taxiway.adb -- This package body retrieves the Oconus_Taxiway records by calling three separate packages (CREATE, BIND and CONVERT) and is a child package to the DM_Airport package. It also interfaces with Open Database Connectivity (ODBC) to allow applications to access data in the database management system (DBMS) using Structured Query Language (SQL).

dm_airport-dm_oconus_taxiway.ads -- This package specification declares the database

record layout to retrieve the Oconus_Taxiway records. It also defines the interface to other programs.

dm_airport-dm_refueling-bind_dm_refueling.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by every data manager to create SQL and ODBC binding.

dm_airport-dm_refueling-convert_to_refueling_data.adb -- This package body is a separate function called by every data manager to assign the retrieved values from the database to the returned variables.

dm_airport-dm_refueling-create_refueling_sql_statement.adb - - This package body is a separate function called by Data Managers to create a SQL statement for a particular data manager based on the given key.

dm_airport-dm_refueling.adb --This package body retrieves the Refueling records by calling three Separate packages (CREATE, BIND and CONVERT) and is a child package to the DM_Airport package. It also interfaces with Open Database Connectivity (ODBC) to allow applications to access data in the database management system (DBMS) using Structured Query Language (SQL).

*dm_airport-dm_refueling.ad*s -- This package specification controls access to the primary runway record.

dm_airport-dm_revetments-create_revetments_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the Revetments data manager based on the user supplied key.

dm_airport-dm_revetments.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_revetments.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_runway-bind_runway.adb -- This package body is a separate procedure which binds the SQL record with ODBC. It is being called by the runway data manager to create SQL and ODBC binding.

dm_airport-dm_runway-convert_to_runway_data.adb -- This package body is a separate function called by the runway data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_runway-create_runway_sql_statement.adb -- This package body is a

separate package function to create an SQL statement for the Runway data manager based on the user supplied key.

dm_airport-dm_runway.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_runway.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_shed-bind_shed.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by the shed data manager to create SQL and ODBC binding.

dm_airport-dm_shed-convert_to_shed_data.adb -- This package body is a separate function called by the shed data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_shed-create_shed_sql_statement.adb -- This package body is a separate package function to create an SQL statement for the Runway data manager based on the user supplied key.

dm_airport-dm_shed.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_shed.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_taxiway.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_taxiway.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_warehouse-bind_dm_warehouse.adb -- This package body is a separate procedure which binds the SQL record with ODBC. This procedure is being called by the warehouse data manager to create SQL and ODBC bindings.

dm_airport-dm_warehouse-convert_to_warehouse_data.adb -- This package body is a separate function called by the warehouse data manager to assign the retrieved values from the database to the return variables.

dm_airport-dm_warehouse-create_warehouse_sql_statement.adb -- This package body is a separate function to create an SQL statement for the warehouse data manager based

on the user supplied key.

dm_airport-dm_warehouse.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-dm_warehouse.ads -- This package specification controls the access to the primary runway record.

dm_airport-dm_weather.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-get_security_information.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_airport-split_be.adb -- This package body provides an SQL search of the Airfields database based on user request.

dm_turnaround.adb -- This package body contains the procedures to clear the report and convert it to a string.

dm_trunaround.ads -- This package builds the type of report selected by the user according to selection criteria. Once a report is created, the report will be displayed on screen and the user will be able to print it.

full_spelling.ads -- This package contains an array that translate codes into full English spelling. The translation is passed back as a string to the calling program.

get_trurnaround_screen_support_info.adb -- This package gets user input from turnaround screen and returns the user selection.

help_panel_dialog_class-interact.adb -- A separate procedure in the Help Panel Dialog Class package which displays the Help Panel and provides the interface to the developer's code. The Help Panel displays the the text box with users manual. This is a Screen Machine artifact, modifed to call other (developer) modules.

help_Panel_dialog_class.adb -- A Package body which defines procedure and functions to perform basic interaction with the Help_Panel. The Help Panel displays a text box with users manual. This is a screen Machine artifact.

help_panel_dialog_class.ads -- Package specification which defines procedures and function to perform basic interaction with the Help_Panel. The Help_Panel displays a text box with users manual.

inset_zeros.adb -- Package body which accomdates for the blank space at the beginning of the variable 'The_Number', which was a number passed in as a string with the help of the integer image function provided in Ada.

main_panel_dialog_class-background.adb -- Package body which defines procedures and function to perform basic interaction with the Main_Panel. This is not a Screen Machine artifact.

main_panel_dialog_class.background.ads -- Package specification which defines procedures and function to perform basic interation with the Main_Panel. This is not a Screen Machine artifact.

main_panel_dialog_class-interact.adb -- A separate procedure in the Main_Panel_Dialog Class Package which displays the Main Panel and provides the interface to the developer's code. The Main Panel displays the main panel. This is a Screen Machine artifact.

main_panel_dialog_class.adb -- Package body which defines procedures and founction to perform basic interaction to the Main_Panel. The Main Panel displays the main panel. This is a Screen Machine artifact.

main_panel_dialog_class.ads -- Package specification which defines procedures and function to perform basic interaction with the Main_Panel. This is a Screen Machine artifact.

make_be.adb -- A function which gets the world area code and installation number id and makes the BE number.

multi_page_report-clear.adb -- A separate procedure which clears the Multi-Page screen report.

multi_page_report-to_string.adb -- A function which formats headers for the Multi-Page report.

multi_page_report.adb -- This package body builds the multipage detail report and the selective data retrieval report from the selection criteria retrieved from the user.

multi_page_report.ads -- This package specification sets up the subroutines for building the multipage detail report and the selective data retrieval report.
odbc-ext.adb -- This package body contains the pragma imports to the drivers.

odbc-ext.ads -- This package specification contains the pragma imports to the drivers.

odbc.ads -- This package specification contains the functions for accessing ODBC.

odbc_utilities.adb -- Package specification containing the utilities for ODBC calls.

odbc_utilities.ads -- Package specification containing the utilities for ODBC calls.

one_line_report-with_coord_radius.adb -- Package body for building the one line report with a coordinate/radius as input.

one_line_report-with_coord_radius.ads -- Package specification for building the one line report with a coordinate/radius as input.

one_page_report.ads -- Package specification for building the one page report.

one_page_report.adb -- Package body for building the one page report.

print_string.adb -- Package body used for debugging. Helps detect unprintable characters.

report_printer.adb -- Package body for printing report from screen to printer.

report_printer.ads -- Package specification for printing report from screen to printer.

screen_machine_housekeeper_package.adb -- Package body that manages all screen machine functions.

screen_machine_housekeeper_package.ads -- Package specification that manages all screen machine functions.

selection_criteria.adb -- Package body that sets up the selection criteria screen and gets user input from the screen.

selection_criteria.ads -- Package specification that sets up the selection criteria screen and gets user input from the screen.

selection_criteria_airfield_name.adb -- Package body that sets up the display for fields on the selection criteria screen for Airfield names.

selection_criteria_airfield_name.ads -- Package specification that sets up the display for fields on the selection criteria screen for basic encyclopedia numbers.

selection_criteria_basic_encyclopedia.adb -- Package body that sets up the display for fields on the selection criteria screen for basic encyclopedia numbers.

selection_criteria_basic_encyclopedia. ads -- Package specification that sets up the

display for fields on the selection criteria screen for Airfield names.

selection_criteria_coordinate_radius-arccos_check.adb -- This separate function returns the ARCCOS function in degrees, but first checks that the argument is not slightly out of range due to round_off error.

selection_criteria_coordinate_radius-calculate_distance_between_in_degrees.adb -- This separate function will calculate the distance in degrees between two points on the globe.

selection_criteria_coordinate_radius-convert_back_to_latitude_type.adb -- This separate function converts from distance_Type to latitude_type.

selection_criteria_coordinate_radius-convert_back_to_longitude_type.adb -- This separate function converts the distance_type to longitude_type.

selection_criteria_coordinate_radius-convert_distance_to_degrees.adb -- This separate function converts the distance from distance_type to degree_type.

selection_criteria_coordinate_radius-convert_lat_to_distance_type.adb -- This separate function will convert latitude to a distance_type.

selection_criteria_coordinate_radius-convert_lat_to_float_type.adb -- This separate function converts the latitude to a float_type.

selection_criteria_coordinate_radius-convert_lon_to_distance_type.adb

selection_criteria_coordinate_radius-convert_lon_to_float_type.adb

selection_criteria_coordinate_radius-determine_distance.adb -- This separate function will calculate the distance in nautical miles between two points on the globe.

selection_criteria_coordinate_radius-determine_lat_longs.adb -- This separate function takes the coordinate point and distance from the point and provide the coordinates repersenting a square around the coordinate which includes more than all coordinates that fall within the distance provided. When coordinate point and distance encompass one of the poles, then minimum and maximum longitude will be repersented by 00E and 00W.

selection_criteria_coordinate_radius-new_geographic_location.adb -- This separate function is taken from Granite Sentry phase II program. NTD A_Tactical_Decision_Aid_Pkg dated 1 November, 1989. This function will provide new location based on given point, course and distance to travel.

selection_criteria_coordinate_radius.adb -- This package body sets up the display for the fields on the selection criteria screen and gets all user input and validates it.

selection_criteria_coordinate_radius.ads -- This package specification controls the access to the coordinate_radius record.

selection_criteria_country_code.adb -- This package specification sets up the display for fields on the selection criteria screen for Country_Codes and receives and validates user input.

selection_criteria_country_code.ads -- This package specification controls the access to the country_code record.

selection_criteria_geoloc.adb -- This package body sets up the display for fields on the selection criteria screen for geolocs and receives and validates the Geoloc input/selected by the user.

selection_criteria_geoloc.ads -- This package body sets up the display for fields on the selection criteria screen for geolocs and receives and validates the Geoloc input/selected by the user.

selection_criteria_icao.adb -- This package body sets up the display for fields on the selection criteria screen for ICAO codes.

*selection_criteria*_icao.ads -- This package specification sets up the display for fields on the selection criteria screen for ICAO codes.

selection_criteria_panel_dialog_class-extras-background.adb -- This package body implements the background processing for the 'Selection_Criteria_Panel' panel.

selection_criteria_panel_dialog_class-extras-background.ads -- This package specification implements the background processing for the 'Selection_Criteria_Panel' panel.

selection_criteria_panel_dialog_class-extras.adb -- This package body adds extra procedures to the selection criteria panel dialog class. These procedures are needed to provide behind-the-scenes processing for the panel.

selection_criteria_panel_dialog_class-extras.ads --This package specification adds extra procedures to the selection criteria panel dialog class. These procedures are needed to provide behind-the-scenes processing for the panel.

selection_criteria_panel_dialog_class-interact.adb -- Screen Machine generated body for

interacting with the user via the “Selection_Criteria_Panel” panel.

selection_criteria_panel_dialog_class.adb -- Screen Machine generated specification for interacting with the user via the “Selection_Criteria_Panel” panel.

selection_criteria_panel_dialog_class.ads -- Screen Machine generated specification for interacting with the user via the “Selection_Criteria_Panel” panel.

selective_panel_dialog_class-interact.adb -- Package body that implements the interactive dialog logic for the “Selective Panel” panel.

selective_panel_dialog_class-process_list_choices.adb -- Package body that sets up the display for the Selective Data Retrieval report.

selective_panel_dialog_class-reset_lists.adb -- Package body that resets the display for the Selective Data Retrieval report.

selective_panel_dialog_class.adb -- Package body generated by Screen Machine that establishes the dialog class for interacting with the user via the “Selective Data_Panel” panel.

selective_panel_dialog_class.ads -- Package specification generated by Screen Machine that establishes the dialog class for interacting with the user via the “Selective_Data_Panel” panel.

storage_manager_sequential.adb -- Package body used for managing storage.

storage_manager_sequential.ads -- Package specification used for managing storage.

string_sequential_unbounded_managed_iterator.adb -- Generic package body used for string manipulation.

string_sequential_unbounded_managed_iterator.ads -- Generic package specification used for string manipulation.

turnaround_panel_dialog_class-interact.adb -- Package body generated by Screen Machine that sets up the dialog logic for the “Turnaround_Panel” panel.

turnaround_panel_dialog_class.adb -- Package body generated by Screen Machine that establishes the dialog class for interacting with the user via the “Turnaround_Panel” panel.

turnaround_panel_dialog_class.ads -- Package specification generated by Screen Machine that establishes the dialog class for interacting with the user via the

"Turnaround_Panel" panel.

turnaround_report.adb -- Package body that builds the turnaround report.

turnaround_report.ads -- Package specification that builds the turnaround report.

turnaround_screen_support-simulation_support-aircraft_capacity.adb -- Package body that gets the largest aircraft capacity from the user for calculating the turnaround time.

turnaround_screen_support-simulation_support-aircraft_capacity.ads -- Package specification that gets the largest aircraft capacity from the user for calculating the turnaround time.

turnaround_screen_support-simulation_support-load_class.adb -- Package body that gets the minimum load class number from the user for calculating the turnaround time.

turnaround_screen_support-simulation_support-load_class.ads -- Package specification that gets the minimum load class number from the user for calculating the turnaround time.

turnaround_screen_support-simulation_support.adb -- Package body that gets the the square feet, ground time, min runway length , and min taxiway width from the user for calculating the turnaround time.

turnaround_screen_support-simulation_support.ads -- Package specification that gets the the square feet, ground time, min runway length , and min taxiway width from the user for calculating the turnaround time.

turnaround_screen_support.ads -- Package specification that sets up the display for user input. Input used to produce the turnaround report.

turnaround_screen_support.adb -- Package body that sets up the display for user input. Input used to produce the turnaround report.

3.4 Changes Installed. The following are the changes to the system when compared with the WWMCCS version of the Airfields system:

(a) The GCCS Airfields system is PC resident with client and server architecture. The current version of Airfields is installed on a Sun Sparc workstation in the GCCS Common Operating Environment (COE). It runs under Sun Solaris 2.3 and database manipulations are handled via the Oracle Standard Query Language (SQL). Sequel Loader is used to load the migrate tables. An Open Database Connectivity (ODBC) package is utilized to link the application to the database. Screen Machine, A

graphical User Interface (GUI) is used at the front end of the system.

(b) The GCCS Airfields system has been re-engineered/re-hosted from a WWMCCS COBOL application to an Ada 95 Programming language, using Oracle 7 as the Relational Database Management System (RDBMS) and SQL.

(c) The GCCS Airfields system complies with GCCS Integration Standards and employs many standards such as the windowing capability and an extensive Help capability to aid the user with system operation.

3.5 Adaptation Data. There is no unique-to-site data contained in this software version.

3.6 Database Installation. The GCCS Database System Administrator (SA) will run the Airfields database segment using the Segment Installer. The new version of Airfields database segment must be available to the SA .

a. Prior to installation, a new ORACLE password must be selected. Verify installation of required segment.

b. Deinstall previous version of the AirField DB SERVER [version #].

c. Install the new version of the AirField DB SERVER [version #]. The database SA will be prompted to enter the directory to hold the ORACLE tablespace AIRFDB and the new password.

d. The following script must be executed by the database SA:
h/AIRFDB/install/airfield_grant_user userid
where userid = UNIX user account of user who runs the Airfields

NOTE: To revoke AIRFIELDS privileges from an existing user, the following script must be executed by the database SA:

h/AIRFDB/install/airfield_revoke_user userid
where userid = UNIX user account of user who runs the Airfields

e. Enter <RETURN> to confirm.

3.7 Application Installation. The GCCS System Administrator (SA) will run the Airfields application segment using the Segment Installer. The Airfields application goes on the application server. The new version of Airfields must be available to the SA . In order to install the application, the following must be accomplished.:

a. Steps for the Initial installation of the Airfield systems follow:

(1) The first time user must obtain the Computer Network Node Name and System Host ID for the system that the application will be run on by keying in the "hostname" and "hostid".

(2) Call the GCCS hotline at DSN 653-8681 or commercial (703) 735-8681 and provide the "hostname" and "hostid" in order to obtain a license file for the system.

(3) The GCCS System Administrator will run the Segment Installer using the Airfields Script Segmentation. If necessary, de-install any previous version of the Airfields application and install the new version.

(4) The SA must switch to the *license* directory of the Airfields application. A file called *license.dat* must be created under the *license* directory using the license key provided by the GCCS hotline. The turnaround time for this action is between 2 and 24 hours.
NOTE: For instructions on how to activate the license file once it has been obtained, see the /h/AirFields/license/license_installation.readme file.

(5) From the command line, execute the following command:
source S99Airfields <ENTER>

(6) Click on the Airfields ICON.

b. Steps for the <u>subsequent installation</u> or updates to the Airfield systems follow:

(1) The *license.dat* file must be saved to a directory outside the Airfields application directories in order to preserve it.

(2) The GCCS System Administrator will run the Segment Installer using the Airfields Script Segmentation. De-install the previous version of the Airfields application and install the new version.

(3) After the installation is complete, copy the *license.dat* file back to the *license* directory.

(4) From the command line, execute the following command:
source S99Airfields <ENTER>

(5) Click on the Airfields ICON.

3.7.1 <u>Security and Privacy</u>. The database is classified Secret/No Foreign Dissemination (SNF).

On-line reports from the retrieval system are marked with the highest classification of data actually reported. Continental U.S.A. (CONUS) data is unclassified. All other airfields can be classified up to SNF. Users are advised to control classified reports properly.

3.8 Possible Problems and Known Errors. There are currently no possible problems or known errors in the software.

SECTION 4. NOTES

4.1 Terms and Abbreviations.

CFOS	Center for Operational Support
COBOL	Common Business Oriented Language
DBMS	Data Base Management System
DDA	Designated Development Agency
DID	Data Item Description
DISA	Defense Information Systems Agency
DMAAC	Defense Mapping Agency Aerospace Center
GCCS	Global Command and Control Systems
GUI	Graphical User Interface
JIEO	Joint Interoperability & Engineering Organization
NOFORN	No Foreign Dissemination
ODBC	Open Database Connectivity
OPR	Office of Primary Responsibility
RDBMS	Relational Database Management System
SDP	Software Development Plan
SNF	Secret/No Foreign [dissemination]
STD	Standard
SUM	Software Users Manual
SVD	Software Version Description
WWMCCS	Worldwide Military Command and Control Systems